KP+ : Fixing Availability Issues on KP Ownership Transfer Protocols

Jorge Munilla

***Abstract*—Ownership Transfer Protocols for RFID allow transferring the rights over a tag from a current owner to a new owner in a secure and private way. Recently, Kapoor and Piramuthu have proposed two schemes which solve most of the security weaknesses detected in previously published protocols. However, this paper reviews this work and points out that such schemes still present some practical and security issues. We then propose some modifications in these protocols that overcome such problems.**

***Index Terms*—RFID, privacy, unlinkability, DoS, forward secrecy, de-synchronization, protocol failure.**

I. Introduction

RADIO Frequency Identification (RFID) is a well established wireless technology for inventory, retail and supply-chain management. However, this technology faces different risks such as lack of privacy or confidentiality, malicious traceability and loss of data integrity, which can only be prevented with the implementation of security mechanisms that take into account its special characteristics: vulnerabilities of radio channel, power-constrained devices, low-cost tags with limited functionalities and data promiscuously transmitted when excited by being in close proximity to the reader [1].

Ownership Transfer Protocols (OTPs) allow the secure transfer of the (digital) ownership of a tag from a current owner to a new owner. Thus, three different roles or entities are always present in an OTP: the item or tag T whose rights are going to be transferred, the seller or Current Owner, who has the initial control of T, and the buyer or New Owner, who will have the control of T when the protocol succeeds. In order to prevent previous owners can access the tag once it has been transferred, two different mechanisms are usually used: the presence of a Trusted Third Party (TTP), which coordinates the transaction, and the assumption of an Isolated Environment (IsE), where, after the private information has been transferred, the new owner can update the keys without being eavesdropped by the previous owner. Both schemes make sense depending on the application [2]. The first provides higher security for strong adversarial scenarios and the second is more appropriate when tags belong to independent authorities or companies.

The first works dealing with Ownership Transfer in the RFID framework were published in 2005 by Molnar et al. [3] and Saito et al. [4]. Recently, Kapoor and Piramuthu have reviewed these and other subsequent proposals [5]–[8] and have proposed two new schemes [9], based on TTP and IsE respectively. A variant of these protocols for multiple tags have also been published [10]. Other OTPs can be found in the literature (e.g. [11]–[15]) but many of them present flaws or vulnerabilities [16], [17]. Thus, in this letter, we review Kapoor and Piramuthu's schemes, which are claimed to be more secure than those currently existing and yet just as lightweight, and we will show that although they address most of the problems encountered in previous proposals, they still raise other practical and security issues which should be corrected. Thus, our goal in this letter is to propose enhanced versions of these protocols, KP+, that with slight modifications, overcome the mentioned problems. We consider that it is important that potential implementers of the prominent KP protocols know the results of the analysis conducted in this paper and hope that it can help in the development of new designs.

J. Munilla is with the Communication Engineering Department, Univ. de Málaga, Spain, 29071. E-mail: munilla@ic.uma.es

Manuscript received xxxx, 20xx; revised December xx, 20xx.

II. Kapoor and Piramuthu's Protocols

These schemes use two keyed encryption (key k) functions: g_k, between the high-level entities, and f_k, between the tag and the other entities; and a secure hash function $H_k()$. In the description of the protocols, $\mathcal{T}$ will stand for the tag which is going to be transferred, and for the sake of simplicity, we will use $\mathcal{R}1$ and $\mathcal{R}2$ to refer to the readers of the current and the new owner respectively.

A. Kapoor and Piramuthu's Protocol with TTP

In the KP protocol *with Trusted Third Party*, $\mathcal{TTP}$ shares static secret keys r_1 and r_2 with $\mathcal{R}1$ and $\mathcal{R}2$ respectively, and a secret key t_i with $\mathcal{T}$, different for each tag. Additionally, $\mathcal{TTP}$ knows the key s_1 that $\mathcal{T}$ currently shares with $\mathcal{R}1$, and it will generate the key s_2 that $\mathcal{T}$ will share with $\mathcal{R}2$. This protocol is accomplished as follows (see Fig. 1):

S.1) Upon receiving an Ownership Transfer Request, $\mathcal{TTP}$ generates a random nonce N_P and a new key s_2, and authenticates itself to $\mathcal{T}$ by sending $f_{(N_P \oplus t_i \oplus s_1)}(s_2)$ along with N_P.

$\mathcal{TTP} \rightarrow \mathcal{T}$: N_P, $f_{(N_P \oplus t_i \oplus s_1)}(s_2)$

S.2) $\mathcal{T}$ checks the received message. If it is correct, $\mathcal{T}$ updates s_1 to s_2, and acknowledges it by generating a random nonce $N_\mathcal{T}$ and using the one-way hash H with this value.

$\mathcal{T} \rightarrow \mathcal{TTP}$: $N_\mathcal{T}$, $H_{(t_i \oplus N_\mathcal{T})}(s_2 \oplus N_P)$

S.3) $\mathcal{TTP}$ informs the current owner ($\mathcal{R}1$) that his privileges are being revoked by sending a value computed with the keyed cryptographic function (along with a simple revoke message).

$\mathcal{TTP} \rightarrow \mathcal{R}1$: $g_{r_1}(s_1)$

S.4) $\mathcal{TTP}$ generates a new random nonce N'_P and sends it and $g_{(r_2 \oplus N'_P)}(s_2 \oplus r_2)$ to $\mathcal{R}2$.

$\mathcal{TTP} \rightarrow \mathcal{R}2$: N'_P, $g_{(r_2 \oplus N'_P)}(s_2 \oplus r_2)$

S.5) The new owner ($\mathcal{R}2$) sends an acknowledgment with the new key value to $\mathcal{TTP}$.

$\mathcal{R}2 \rightarrow \mathcal{TTP}$: $H_{r_2}(s_2 \oplus N'_P)$

S.6) Furthermore, $\mathcal{R}2$ generates a random nonce $N_{\mathcal{R}2}$ and sends it to $\mathcal{T}$ encrypted by using the key s_2.

$\mathcal{R}2 \rightarrow \mathcal{T}$: $N_{\mathcal{R}2}$, $f_{s_2}(N_{\mathcal{R}2})$

S.7) $\mathcal{T}$, to acknowledge that the message is correct, sends the hashed value of a new random nonce $N'_{\mathcal{T}}$ along with $N_{\mathcal{R}2}$ and s_2.

$\mathcal{T} \rightarrow \mathcal{R}2$: $N'_{\mathcal{T}}$, $H_{(N'_{\mathcal{T}} \oplus s_2)}(N_{\mathcal{R}2} \oplus s_2)$

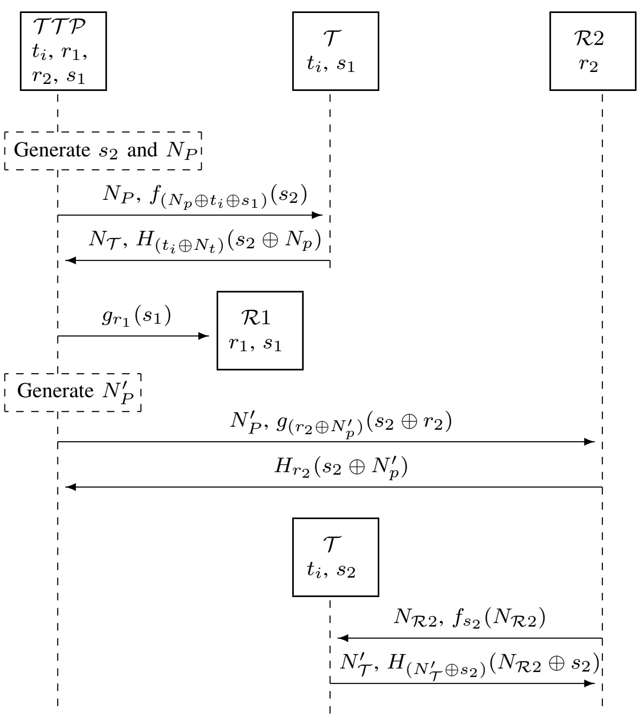

Fig. 1. Kapoor and Piramuthu's OTP with TTP

B. Kapoor and Piramuthu's Protocol without TTP

This assumes a secure channel between $\mathcal{R}1$ and $\mathcal{R}2$. The protocol is presented in Figure 2 and described below.

S.1) Upon receiving a request for Ownership Transfer, $\mathcal{R}1$ generates a fresh random number $N_{\mathcal{R}1}$, computes $N_{\mathcal{R}1} \oplus s_1$, where s_1 is the key that $\mathcal{R}1$ shares with $\mathcal{T}$, and sends the result to $\mathcal{R}2$ over a secure channel.

$\mathcal{R}1 \Rightarrow \mathcal{R}2$: $N_{\mathcal{R}1} \oplus s_1$

S.2) $\mathcal{R}1$ sends the same information to $\mathcal{T}$ but encrypted with s_1.

$\mathcal{R}1 \rightarrow \mathcal{T}$: $f_{s_1}(N_{\mathcal{R}1} \oplus s_1)$

S.3) $\mathcal{T}$ generates two fresh random numbers: $N_{\mathcal{T}}$ and $N'_{\mathcal{T}}$; and computes the value $N = N_{\mathcal{R}1} \oplus N_{\mathcal{T}}$. Then, $\mathcal{T}$ randomly $flip$s one bit in N, creating N'. $\mathcal{T}$ sends the following messages to $\mathcal{R}2$:

$\mathcal{T} \rightarrow \mathcal{R}2$: $N_{\mathcal{T}} \oplus s_1$, $N'_{\mathcal{T}}$, $f_{(N' \oplus N'_{\mathcal{T}})}(N' \oplus N'_{\mathcal{T}})$, $H_{(N' \oplus N'_{\mathcal{T}})}(N' \oplus N'_{\mathcal{T}})$

S.4) Now, both $\mathcal{T}$ and $\mathcal{R}2$ know N. Knowing N, $\mathcal{R}2$ uses a brute force technique on $f_{(N' \oplus N'_{\mathcal{T}})}(N' \oplus N'_{\mathcal{T}})$ to determine N', and checks the computed result with the hash value $H_{(N' \oplus N'_{\mathcal{T}})}((N' \oplus N'_{\mathcal{T}})$. Then, $\mathcal{R}2$ generates a new key s_2 and sends the following message to $\mathcal{T}$:

$\mathcal{R}2 \rightarrow \mathcal{T}$: $f_{N'}(N' \oplus s_2)$

S.5) The previous step is repeated after a predetermined time period until $\mathcal{T}$ acknowledges receipt of the new key, by using it with the hash function.

$\mathcal{T} \rightarrow \mathcal{R}2$: $H_{s_2}(N' \oplus s_2)$

S.6) $\mathcal{R}2$ sends $f_{s_2}(N' \oplus s_2)$ to acknowledge receipt of the message in the previous step.

$\mathcal{R}2 \rightarrow \mathcal{T}$: $f_{s_2}(N' \oplus s_2)$

If the tag does not receive this within a predetermined amount of time, the process is repeated from the beginning.

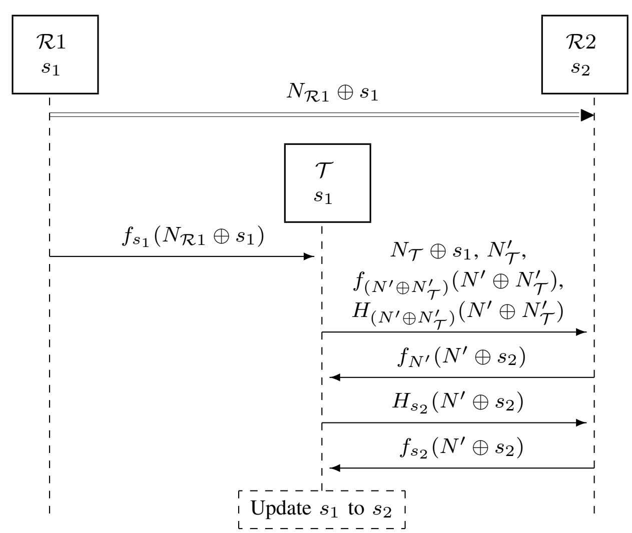

Fig. 2. Kapoor and Piramuthu's OTP without TTP.

III. Cryptanalysis

A. Desynchronization attack on KP Protocol with TTP

According to the authors, the message $\{N_P, f_{(N_p \oplus t_i \oplus s_1)}(s_2)\}$, in Step 1, authenticates the TTP to the tag, which updates s_1 to s_2. However, this is not correct and an adversary can send forged messages that make $\mathcal{T}$ update its key to a fake value $s_{\mathcal{A}}$, causing desynchronization.

Proof. Let $\mathcal{A}$ be an adversary that, impersonating $\mathcal{TTP}$, sends any two values "$N_{\mathcal{A}}$, $F_{\mathcal{A}}$" to $\mathcal{T}$ in Step 1. Then, $\mathcal{T}$ will decrypt $F_{\mathcal{A}}$ and update s_1 to s_a, with $s_{\mathcal{A}}$=$f^{-1}_{(N_{\mathcal{A}} \oplus t_i \oplus s_1)}(F_{\mathcal{A}})$. □

B. DoS attacks on KP Protocol without TTP

The values sent by $\mathcal{T}$ in Step 3,

$$\{N_{\mathcal{T}} \oplus s_1, N'_{\mathcal{T}}, f_{(N' \oplus N'_{\mathcal{T}})}(N' \oplus N'_{\mathcal{T}}), H_{(N' \oplus N'_{\mathcal{T}})}(N' \oplus N'_{\mathcal{T}})\}$$

does not provide integrity on N', which causes that an adversary can modify intercepted messages to generate new forged messages that will be accepted by $\mathcal{R}2$, causing the protocol to go into a endless loop.

Proof. Let $A = N_{\mathcal{T}} \oplus s_1$, $B = N'_{\mathcal{T}}$, $C = f_{(N' \oplus N'_{\mathcal{T}})}(N' \oplus N'_{\mathcal{T}})$ and $D = H_{(N' \oplus N'_{\mathcal{T}})}(N' \oplus N'_{\mathcal{T}})$ be valid messages intercepted by an adversary $\mathcal{A}$ in Step 3. For any new value $A_{\mathcal{A}}$, the adversary generates and sends (Man In the Middle Attack) a new set of values:

$$\{A_{\mathcal{A}}, B_{\mathcal{A}} = B \oplus A \oplus A_{\mathcal{A}}, C, D\}.$$

These values will be accepted by $\mathcal{R}2$, since $A_{\mathcal{A}} \oplus s_1 \oplus B_{\mathcal{A}} = N_{\mathcal{T}} \oplus N'_{\mathcal{T}}$. Thus, the protocol continues normally but $\mathcal{R}2$ computes an incorrect value $N'_{\mathcal{A}} = flip(N_{\mathcal{R}1} \oplus N_{\mathcal{T}} \oplus \Delta) = flip(N_{\mathcal{R}1} \oplus N_{\mathcal{T}}) \oplus \Delta = N' \oplus \Delta$, with $\Delta = A_{\mathcal{A}} \oplus A$. As a result, $\mathcal{T}$ and $\mathcal{R}2$ assume different values for N' and, according to the description of the protocol, steps 3 will be repeated indefinitely (because tag cannot acknowledge receipt of the new key in Step 4). □

IV. KP+

A. KP+ with TTP

The availability problem of KP protocol described in Section III-A is caused because the flow in Step 1 does no authenticate $\mathcal{TTP}$ to $\mathcal{T}$. A new flow whose computation includes session (fresh) randomness provided by the tag must be added, and only after this flow is checked must the tag update its key. We propose this flow takes place between Step 2 and Step 3, while the rest of the protocol remains unchanged (see Fig. 3):

S.2-3) Upon receiving the message $H_{(t_i \oplus N_{\mathcal{T}})}(s_2 \oplus N_P)$, that authenticates $\mathcal{T}$, and $N_{\mathcal{T}}$ that provides randomness for this session, $\mathcal{TTP}$ computes and replies with $f_{s_2}(s_1 \oplus N_{\mathcal{T}})$.

$\mathcal{TTP} \rightarrow \mathcal{T}$: $f_{s_2}(s_1 \oplus N_{\mathcal{T}})$

$\mathcal{T}$ checks if this is correct and if so, updates s_1 to s_2. The rest of the protocol remains unchanged. The computation of $f_{s_2}(s_1 \oplus N_{\mathcal{T}})$ proves the authorship of $\mathcal{TTP}$ since it requires the knowledge of s_1 and s_2, and the use of $N_{\mathcal{T}}$ guarantees its participation in this particular session (preventing replay attacks). Note also that the option that $\mathcal{T}$ keeps s_1, without updating to s_2 until Step 6, when it receives the confirmation from $\mathcal{R}2$, would not prevent replay attacks with messages (exchanged in Step 1 and Step 6) from interrupted (unsuccessful) sessions.

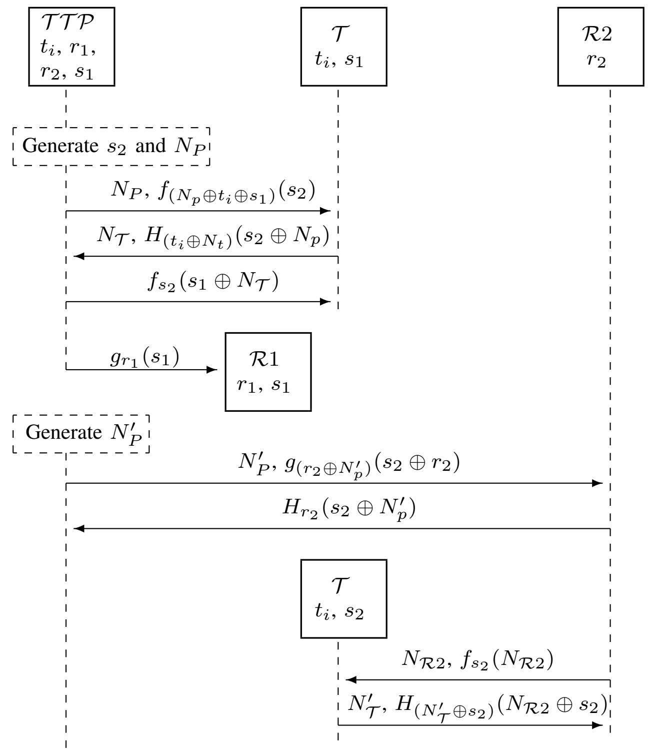

Fig. 3. KP+ with TTP

B. KP+ without TTP

The DoS problem of KP without TTP (Section III-B) is solved by guaranteeing the integrity of $N'_{\mathcal{T}}$ so that it cannot be modified by the adversary without affecting the validity of the other encrypted values. Thus, we propose here to change Step 3 as follows (note that $N'_{\mathcal{T}}$ is not involved in any other flow):

S.3) Upon receiving $f_{s_1}(N_{\mathcal{R}1} \oplus s_1)$ from $\mathcal{R}1$, $\mathcal{T}$ generates two random numbers $N_{\mathcal{T}}$ and $N'_{\mathcal{T}}$, and computes $N = N_{\mathcal{T}} \oplus N_{\mathcal{R}1}$. Then, $\mathcal{T}$ randomly $flip$s one bit in N, creating N' and use it to compute $f_{N'}(N'_{\mathcal{T}})$ and $H_{N'}(N'_{\mathcal{T}})$. Then, $\mathcal{T}$sends to $\mathcal{R}2$ the following message:

$\mathcal{T} \rightarrow \mathcal{R}2$: $N_{\mathcal{T}} \oplus s_1$, $N'_{\mathcal{T}}$, $f_{N'}(N'_{\mathcal{T}})$, $H_{N'}(N'_{\mathcal{T}})$

This new flow is simpler than the original and avoids that new valid fake messages can be generated. If $N_{\mathcal{T}} \oplus s_1$ and/or $N'_{\mathcal{T}}$ are modified, new values $f_N(N'_{\mathcal{T}})$ and $H_{N'}(N'_{\mathcal{T}})$ must be computed; i.e. previously computed values cannot be reused.

V. CONCLUSION

We have proven that the two protocols recently defined by Kapoor and Piramuthu suffer from flaws in their design that allow attackers to break the regular behavior of the system by means of desynchronization or denegation of service attacks. This fact is of vital importance in ownership transfer protocols because they are closely related to commercial transactions. Since these protocols were originally designed to overcome security weaknesses of their predecessors, we have proposed modifications in order to fix the problems.

ACKNOWLEDGMENT

Acknowledgments

REFERENCES

[1] K. Finkenzeller, *RFID Handbook : Fundamentals and Applications in Contactless Smart Cards and Identification*, 2nd ed. John Wiley & Sons, May 2003.

[2] A. Fernandez-Mir, R. Trujillo-Rasua, J. Castella-Roca, and J. Domingo-Ferrer, "A Scalable RFID Authentication Protocol Supporting Ownership Transfer and Controlled Delegation." in *RFIDSec*, ser. Lecture Notes in Computer Science, vol. 7055. Springer, 2011, pp. 147–162.

[3] D. Molnar, A. Soppera, and D. Wagner, "A Scalable, Delegatable Pseudonym Protocol Enabling Ownership Transfer of RFID Tags," in *Workshop on RFID Security and Light-Weight Crypto*, Graz, Austria, July 2005.

[4] J. Saito, K. Imamoto, and K. Sakurai, "Reassignment Scheme of an RFID Tag's Key for Owner Transfer," in *EUC Workshops*, ser. Lecture Notes in Computer Science, T. Enokido, L. Yan, B. Xiao, D. Kim, Y.-S. Dai, and L. T. Yang, Eds., vol. 3823. Springer, 2005, pp. 1303–1312.

[5] K. Osaka, T. Takagi, K. Yamazaki, and O. Takahashi, ""an efficient and secure rfid security method with ownership transfer"," in *Proc. Int. Conf. Comput. Intell. Security*, ser. LNAI 4456, 2007, pp. 778–787.

[6] Y. Seo, T. Asano, H. Lee, and K. Kim, ""a lightweight protocol enabling ownership transfer and granular data access of rfid tags"," in *Proc. Symp. Cryptogr. Inf. Security*, 2007, pp. 1–7.

[7] K. H. S. Sabaragamu Koralalage, S. M. Reza, J. Miura, Y. Goto, and J. Cheng, "Pop method: An approach to enhance the security and privacy of rfid systems used in product lifecycle with an anonymous ownership transferring mechanism," in *Proceedings of the 2007 ACM Symposium on Applied Computing*, ser. SAC '07. New York, NY, USA: ACM, 2007, pp. 270–275. [Online]. Available: http://doi.acm.org/10.1145/1244002.1244069

[8] H. Lei and T. Cao, ""rfid protocol enabling ownership transfer to protect against traceability and dos attacks"," in *Proc. 1st Int. Symp. Data, Privcacy E-Commerce*, 2007, pp. 508–510.

[9] G. Kapoor and S. Piramuthu, "Single RFID Tag Ownership Transfer Protocols," *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, vol. 42, no. 2, pp. 164–173, 2012.

[10] G. Kapoor, W. Zhou, and S. Piramuthu, "Multi-tag and Multi-owner RFID Ownership Transfer in Supply Chains," *Decision Support Systems*, vol. 52, no. 1, pp. 258–270, 2011.

[11] C.-L. Chen, Y.-Y. Chen, Y.-C. Huang, C.-S. Liu, C.-I. Lin, and T.-F. Shih, "Anti-counterfeit ownership transfer protocol for low cost rfid system," *W. Trans. on Comp.*, vol. 7, no. 8, pp. 1149–1158, Aug. 2008. [Online]. Available: http://dl.acm.org/citation.cfm?id=1457999.1458003

[12] T. Dimitriou, "rfiddot: Rfid delegation and ownership transfer made simple," in *Proceedings of the 4th International Conference on Security and Privacy in Communication Netowrks*, ser. SecureComm '08. New York, NY, USA: ACM, 2008, pp. 34:1–34:8. [Online]. Available: http://doi.acm.org/10.1145/1460877.1460921

[13] P. Jäppinen and H. Hämäläinen, "Enhanced rfid security method with ownership transfer," in *Proceedings of the 2008 International Conference on Computational Intelligence and Security - Volume 02*, ser. CIS '08. Washington, DC, USA: IEEE Computer Society, 2008, pp. 382–385. [Online]. Available: http://dx.doi.org/10.1109/CIS.2008.26

[14] C.-H. Wang and S. Chin, "A new rfid authentication protocol with ownership transfer in an insecure communication environment," in *Proceedings of the 2009 Ninth International Conference on Hybrid Intelligent Systems - Volume 01*, ser. HIS '09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 486–491. [Online]. Available: http://dx.doi.org/10.1109/HIS.2009.100

[15] C.-L. Chen and C.-F. Chien, "An ownership transfer scheme using mobile rfids." *Wireless Personal Communications*, vol. 68, no. 3, pp. 1093–1119, 2013.

[16] G. Kapoor and S. Piramuthu, "Vulnerabilities in some recently proposed rfid ownership transfer protocols." in *NetCoM*. IEEE Computer Society, 2009, pp. 354–357.

[17] J. Munilla, F. Guo, and W. Susilo, "Cryptanalaysis of an EPCC1G2 Standard Compliant Ownership Transfer Scheme." *Wireless Personal Communications*, vol. 73, no. 73, pp. 245–258, 2013.